

Uma infra-estrutura para publicação e descoberta de serviços de monitoramento de rede

Herbert M.Souza¹, Rafael Costa¹, José A. Suruagy¹,
Leobino Sampaio^{1,2}, Paulo F. Cunha²

¹Núcleo de Pesquisa em Redes e Computação (NUPERC)
Universidade Salvador (UNIFACS) – Salvador – BA – Brasil

²Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE) - Recife - PE - Brasil

{herbert.souza, rafael.costa, suruagy}@unifacs.br, {lns, prfc}@ufpe.br

Abstract. *Network monitoring environments have been adopting Web Services for distributing their services functionalities and solving communication problems, as well as the discovery and publication technology to solve the location and access problems. This paper proposes a discovery and publication infrastructure geared toward network monitoring services. It is presented as contribution an UDDI data model related to measurement services, the implementation of a new Lookup Service type with UDDI technology, as well as comparative tests with the previous version that is based in a XML database.*

Resumo. *Os ambientes de medições em redes de computadores têm sido implementados com o uso de Serviços Web como forma de distribuir as funcionalidades de suas aplicações e resolver problemas de comunicação, e da tecnologia de publicação e descoberta para sanar o problema de localização e acesso aos serviços. Este trabalho propõe uma infra-estrutura de publicação e descoberta voltada para serviços de monitoramento de redes de computadores. Essa proposta traz também como contribuição um modelo de dados para a tecnologia UDDI voltado para serviços de medição, a implementação de um novo tipo de Lookup Service o qual utiliza a tecnologia UDDI e testes comparativos com a versão anterior que usa base de dados XML.*

1. Introdução

Nos últimos anos o monitoramento das redes de computadores tem sido fortemente influenciado pela tendência de utilização de infra-estruturas [Hanemann et al. 2005, Sampaio et al. 2006] que sejam capazes de operar em redes de padrões abertos como a Internet. Esta tendência se dá justamente pelo fato das atuais aplicações de redes estarem migrando os seus escopos de funcionamento, passando a operar em ambientes multi-domínio. O desenvolvimento de ambientes de monitoramento que englobam múltiplos domínios administrativos geralmente requer a implementação de uma camada de *middleware* que fornece um conjunto de serviços que fazem com que as aplicações usuárias abstraíam os detalhes específicos das ferramentas e técnicas de medição. Atualmente, tais serviços são desenvolvidos e disponibilizados por meio de Serviços Web (do inglês *Web Services*) [Gudgin et al. 2001].

A utilização de Serviços Web basicamente envolve um provedor de serviços, que disponibiliza as informações técnicas disponíveis através de interfaces descritas em documentos WSDL (*Web Services Description Language*) [Roy and Ramanujan 2001], e pelo menos um consumidor dos serviços prestados, que através das descrições disponibilizadas pelo provedor, obtém as informações técnicas de como acessá-los. No entanto, a fim de se ter um maior dinamismo nestas soluções, geralmente é necessário utilizar os três componentes da arquitetura orientada a serviços (SOA) [Papazoglou 2003]. Ou seja, não somente o consumidor e provedor de serviços como também a infraestrutura de publicação e descoberta. Essas infra-estruturas geralmente são implementadas através de tecnologias como UDDI (*Universal Description Discovery and Integration*) [Curbera et al. 2002] e ebXML (*Electronic Business using eXtensible Markup Language*) [Berwanger et al. 2001].

As infra-estruturas de publicação e descoberta de serviços permitem que um serviço seja pesquisado de acordo com informações não somente técnicas, mas também através de informações com maior valor semântico e relacionadas ao domínio em questão. Esta é uma vantagem que se torna ainda mais evidente no universo dos serviços de monitoramento de redes, uma vez que existe uma variedade muito grande de informações sobre os mesmos e seus tipos. Ou seja, existe um conjunto de serviços relacionados à realização dos testes, topologia da rede, autenticação, autorização, controle no uso dos recursos, armazenamento das medidas coletadas, transformação dos dados, etc.

Com essa grande variedade de Serviços Web que fazem parte das infra-estruturas de monitoramento, a necessidade de um dispositivo que armazene as informações sobre os Serviços Web se torna ainda mais imprescindível. Esse armazenamento deve poder refletir as informações de um modelo orientado a mensagens, que se baseia na troca de mensagens entre o consumidor do serviço e o provedor, sendo que várias informações podem estar contidas dentro dessas mensagens. Desse modo, a dinamização e flexibilidade no uso do serviço de publicação e descoberta ajudam tanto na sua disseminação como na dos Serviços Web, pois outros componentes que desejem fazer o uso dos mesmos só precisam implementar uma interface cliente que envie mensagens de requisição para o serviço de publicação e descoberta e receba suas respectivas respostas. Esse tipo de abordagem ajuda na transferência de informações, pois o que trafega entre o cliente e o serviço são apenas documentos XML, os quais podem trazer quaisquer tipos de informação, diferentemente do escopo tradicional via *Remote Procedure Call* - RPC [Roy and Ramanujan 2001], onde as informações são trocadas via parâmetros.

Tecnologias como a UDDI podem ser usadas de forma genérica, já que sua concepção é voltada para a publicação e descoberta de Serviços Web de uma forma geral. No entanto, a carência de referências na literatura que apresentem trabalhos relacionados ao uso do UDDI fora do escopo da área de comércio eletrônico faz com que o seu uso não seja considerado em outras áreas. Por tais motivos, durante o desenvolvimento do *framework* de medição perfSONAR (*Performance focused Service Oriented Network monitoring ARchitecture*) [Hanemann et al. 2005], um novo Serviço de Publicação e Descoberta chamado de *Lookup Service* (LS) foi desenvolvido e com isso todos os aspectos relacionados à comunicação entre as entidades, controle na inserção de informações, níveis de visões dos dados, entre outros, tiveram que ser repensados. Contudo o *Lookup Service* é uma implementação de um serviço de publicação com especificações próprias, que usa

troca de mensagens XML padronizadas e que armazena as informações de publicação dos Serviços Web em base de dados XML. A abordagem de se usar uma base de dados XML facilita o armazenamento, já que as informações sobre os Serviços Web já estão no formato XML. Por outro lado, essa utilização implica em alguns problemas como baixo desempenho e complexidade na manipulação das mensagens, que também serão discutidos neste artigo.

Ao contrário do LS do perfSONAR, o trabalho com objetos previstos nas implementações do UDDI facilita a manipulação das informações, pois com o uso de métodos de acesso, a ação de recuperar e inserir informações se torna mais fácil e ágil. Uma solução para alguns dos problemas de desempenho e manipulação do LS é a união dessa implementação com a tecnologia UDDI que já possui uma estrutura voltada para a Publicação e Descoberta, só faltando o escopo voltado para serviços de monitoramento. O pretendido neste artigo também é demonstrar essa integração entre o UDDI e o LS do perfSONAR, o que gerou uma nova versão do serviço de *Lookup*. Com a integração entre o UDDI e o serviço de *Lookup* além de tirar de cena o uso da base de dados XML, recursos do UDDI como: índices, taxonomias e componentes irão melhorar consideravelmente o desempenho e a manipulação das informações. Este artigo também apresenta a união dessas duas tecnologias e suas implicações.

Devido ao fato de não existirem estudos que envolvam a tecnologia UDDI no contexto de medições, o trabalho apresentado nesse artigo traz importantes contribuições. Portanto, além de uma nova abordagem para a implantação de uma infra-estrutura de Publicação e Descoberta de serviços, será demonstrada também uma metodologia para construção do serviço e o uso da tecnologia UDDI em um escopo diferente do de comércio eletrônico. Este artigo está organizado da seguinte forma: na seção 2 são apresentados alguns trabalhos relacionados ao contexto do desenvolvimento do serviço de Publicação e Descoberta, e em seguida (seção 3) o modelo de dados proposto para a construção do serviço. Logo após, na seção 4, é feito o detalhamento da implementação realizada e, por fim, nas seções 5 e 6, respectivamente, são apresentados os testes realizados, as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

O desenvolvimento e implantação de infra-estruturas para Publicação e Descoberta de Serviços envolvem o uso de tecnologias que implementam especificações e que facilitam o armazenamento e a busca por informações dos serviços cadastrados. Dentre essas especificações conhecidas destaca-se a UDDI, a qual é implementada por diversas corporações tais como a *Apache Software Foundation*¹ que desenvolveu o JUDDI². Além disso, destacam-se também outros desenvolvimentos que seguem uma especificação própria, tais como o realizado no pertSONAR. Nas subseções seguintes serão detalhadas cada uma dessas iniciativas.

¹Apache Software Foundation - <http://www.apache.org>

²JUDDI - Uma implementação Java da especificação UDDI - <http://ws.apache.org/juddi>

2.1. Especificações de infra-estruturas de Publicação e Descoberta de Serviços

2.1.1. ebXML

O foco do ebXML é criar uma base eletrônica de informações sobre serviços de comércio eletrônico, onde serviços podem achar outros, negociar uns com os outros e realizar a transação. Todas essas operações são executadas automaticamente através da troca de mensagens XML. A especificação ebXML descreve uma maneira de definir os processos para uma negociação entre serviços e as mensagens que são usadas para realizar esse processo, sendo que essas definições podem ser feitas com o uso de um registro público.

A especificação 1.0 do serviço ebXML define um formato e um protocolo para a troca de mensagens para o comércio eletrônico, envolvendo questões como segurança e confiabilidade. O serviço de mensagens do ebXML está definido como um grupo de camadas com base no *Simple Object Access Protocol* (SOAP) [Gudgin et al. 2003] e estende esse protocolo por causa das questões de segurança e confiabilidade necessárias para suportar qualquer tipo *e-business* (comércio eletrônico). Um registro ebXML deve suportar as necessidades de informação de clientes que pretendem usá-lo e fornecer um conjunto de serviços que permita compartilhar informações (mantidas em um repositório) entre os clientes que participem de transações.

Como essa especificação foi concebida visando os serviços de comércio eletrônico, o uso em outro contexto implicaria em uma adaptação da mesma. Existe ainda o fato de que uma adaptação pode implicar em uma reformulação na estrutura ou até mesmo na implementação, sendo um esforço muito custoso e inviável, dada a existência de outras soluções capazes de solucionar o problema.

2.1.2. UDDI

A UDDI é uma iniciativa que foi implantada como projeto de pesquisa por várias companhias interessadas em discutir questões relacionadas à integração entre sistemas de computação. Essa tecnologia provê uma especificação para um serviço de publicação e descoberta de Serviços Web, disponibilizando APIs (*Application Programming Interfaces*) que podem ser utilizadas por diversos sistemas para interagir com o serviço e acessar as funcionalidades do registro UDDI [Apte et al. 2002]. De forma resumida, a UDDI é uma especificação que contém as definições de um serviço para armazenar e prover informações (principalmente de acesso) sobre Serviços Web, seguindo uma especificação em comum, com o objetivo de prover um Serviço de Publicação e Descoberta.

As APIs do UDDI são Serviços Web que recebem mensagens XML definidas pela especificação do consórcio UDDI [Curbera et al. 2002], sendo que essas mensagens seguem padrões que determinam a estrutura de dados na base de dados de um registro UDDI. Os principais componentes da estrutura de dados do UDDI podem ser vistos na Figura 1.

O objetivo do UDDI é esquematizar as informações de acesso ao serviço de modo que a pesquisa do mesmo seja otimizada e possa ser realizada de maneira simples e rápida. Contudo, o UDDI trabalha com esquemas de caracterização do serviço, que podem ser usados como índices para agilizar e facilitar as buscas.

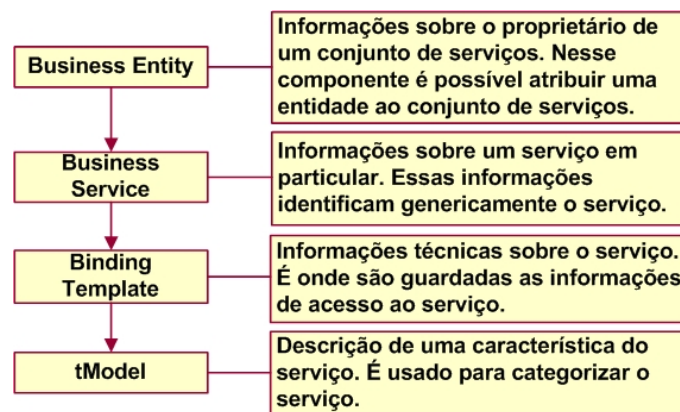


Figura 1. Componentes da tecnologia UDDI.

Outra grande vantagem da tecnologia UDDI é o uso de um componente chamado tModel [Ehnebuske et al. 2002]. O tModel é um índice ao qual os outros componentes do UDDI podem referenciar para fornecer suas funcionalidades, como em um esquema que define um grupo de serviços, ou para a categorização dos mesmos, possibilitando assim a busca por características específicas de cada um. Além disso, o tModel pode fazer referência a outros componentes da estrutura do UDDI e principalmente a outros tModels, relacionando as suas características e refinando ainda mais as buscas. O tModel agiliza a consulta aos serviços e ajuda no relacionamento dos mesmos possibilitando agrupar características.

Com o estudo da viabilidade sobre a tecnologia de Publicação e Descoberta UDDI e ebXML para o uso em serviços de medição, chegou-se à conclusão de que o uso do ebXML não seria viável, já que o mesmo é concebido para serviços de comércio eletrônico, o que implica numa abordagem voltada para a questão do relacionamento entre Serviços Web [Apte et al. 2002] com pouco foco na disponibilização de informações sobre serviços. Essas características de relacionamento entre serviços são pertinentes aos Serviços Web de comércio eletrônico, pelo fato da utilização de vários outros para compor uma solução final para o cliente como: consultas de preço, consultas de informações sobre o cliente, validação de informações, etc.

Por outro lado, a tecnologia UDDI é baseada na criação de um registro padronizado de informações sobre serviços web que pode ser usado de diversas maneiras dependendo do contexto. Como o escopo deste trabalho é armazenar informações sobre serviços e facilitar as buscas para os usuários, a tecnologia UDDI se encaixou perfeitamente devido à possibilidade de uso em vários contextos, como o de medições.

2.2. Serviço de Lookup do perfSONAR

O perfSONAR consiste em um protótipo de uma arquitetura orientada a serviços de monitoramento de redes onde foi desenvolvido um novo serviço de Publicação e Descoberta denominado *Lookup Service* (LS). O LS interage com outros componentes da arquitetura através de Serviços Web via troca de mensagens XML predefinidas. A versão 1.0 desse sistema suporta a publicação, descoberta e alteração das informações sobre os Serviços Web. O sistema utiliza para o armazenamento das informações sobre os serviços, uma base de dados eXist [Meier 2002] (XML). Resumidamente, o sistema funciona da se-

guinte forma: o cliente envia uma requisição em um documento XML para o LS, que por sua vez, possui um Serviço Web genérico, o qual recebe essa mensagem e posteriormente processa, realizando a ação de acordo com o que está contido no documento. Essa ação pode ser a inserção das informações sobre um serviço na base XML (registro) ou a consulta ou a alteração de um serviço. Por fim, após realizar a ação, o serviço envia uma mensagem ao cliente com o resultado da sua requisição.

O LS do perfSONAR é bastante flexível, pois como a troca de mensagens é feita através de documentos XML, a interação com a base de dados se torna mais fácil. Por exemplo, numa requisição de registro de um serviço, o cliente envia o pedido contendo todas as informações para a tarefa e o LS, por sua vez não precisa tratá-las, já que as mesmas são pré-definidas por um esquema e podem ser inseridas na base de dados da mesma forma que foram enviadas. Esse tipo de abordagem facilita a manipulação dos documentos XML, no entanto, dificulta quando é necessária a manipulação dos dados, devido à necessidade de se descobrir e tratar cada uma das informações contidas na mensagem.

O LS possui pontos negativos em vários aspectos, como no armazenamento das informações sobre os serviços, que é feito numa base de dados XML, cujo desempenho é prejudicado pela inexistência de artifícios como o de indexação de uma base relacional. Outro ponto negativo é o fato da não utilização de objetos para a manipulação dos dados, pois o LS assume que as mensagens recebidas devem seguir um padrão pré-definido e qualquer manipulação das informações contidas nessas mensagens devem ser recuperadas através de um *parser*. Outro ponto negativo é o fato da tecnologia ainda estar em desenvolvimento e por esse motivo possuir várias falhas.

3. Modelos de dados para medições

Para construir uma infra-estrutura de Publicação e Descoberta para serviços de medição é necessário definir um modelo de dados de acordo com as necessidades de consulta a serem realizadas na infra-estrutura. Neste trabalho, o modelo de dados foi criado sob a perspectiva de utilização do UDDI, em que foram mapeadas as informações de monitoramento aos seus principais componentes (*Business Service, Binding Template, tModel e etc.*) [Apte et al. 2002] e funcionalidades, levando em consideração também a demanda de informações dos usuários de serviços de medição. Esse modelo de dados serve para distribuir as informações dos serviços sobre os componentes do UDDI visando uma otimização na inserção e consulta dessas informações.

No UDDI, os modelos de dados dão suporte às taxonomias, que fazem a categorização de Serviços Web através de características específicas de um determinado serviço ou de um grupo deles e são representadas pelo componente tModel que pode ser usado como índice para agilizar a pesquisa. Dessa forma, um serviço que usa uma determinada taxonomia possui referências a determinados tModels que correspondem à característica desejada, podendo a busca ser feita por serviços que a referenciam. Um exemplo de taxonomia que pode ser citada é a ISO 3166³, que normalmente é usada para a categorização geográfica, atribuindo códigos para determinadas regiões.

No domínio de monitoramento de redes já existem iniciativas que direcionam esforços no sentido de identificar uma taxonomia que seja um consenso para a maioria

³<http://www.iso.ch/iso/en/prods-services/iso3166ma/index.html>

dos especialistas na área. Dentre as atuais iniciativas, merece destaque o trabalho realizado pelo grupo NMWG [Lowekamp et al. 2003] do *Global Grid Fórum* (GGF)⁴ que procura nomear algumas características para os dados de medições para que possa identificá-los e facilitar o acesso aos mesmos. Este esforço contribui significativamente com o desenvolvimento de uma infra-estrutura de Publicação e Descoberta para serviços de medição, pois os termos identificados acabam se tornando índices que podem ser usados na descoberta dos serviços. A nomenclatura proposta pelo NMWG ainda auxilia na criação do modelo de dados, já que para que o mesmo seja criado, a forma como um cliente busca por informações sobre os serviços deve ser visualizada.

Para o modelo de dados proposto, inicialmente foi definida uma estrutura que contempla as informações básicas de acesso e algumas características específicas demandadas pelos usuários, tais como: nome do serviço, URI (*Universal Resource Identifiers*) do ponto de acesso, localização, ferramenta de medição utilizada e métrica fornecida.

Com a necessidade de seguir um padrão internacional e assim tornar o Serviço de Publicação e Descoberta interoperável com outras infra-estruturas, foi necessário adaptá-lo a um modelo orientado a mensagens XML, fazendo uso do conhecimento já adquirido no LS do perfSONAR no que diz respeito à troca e padronização de mensagens para interface com o mundo exterior. A solução foi a reformulação do modelo de dados no UDDI de acordo com o padrão de mensagem também definido pelo NMWG, que divide a mesma em dois objetos (*Metadata e Data*) e a utilização do esquema de mensagens. O objeto *Metadata* é responsável por armazenar informações genéricas que são comuns a todos os serviços como: nome, ponto de acesso, descrição e tipo, enquanto o objeto *Data* possui informações específicas do serviço, que pode variar de acordo com a estrutura do mesmo. Nesse artigo serão utilizados como exemplo os Serviços Web disponibilizados pelo módulo de armazenamento de dados de utilização dos enlaces da infra-estrutura de medição perfSONAR. Nesse caso, o objeto *Data* armazena as informações referentes às interfaces dos dispositivos onde são coletados os dados de utilização, ou seja, cada objeto irá armazenar as informações sobre cada interface em que estão sendo coletados os dados de utilização.

Na Figura 2 é possível ver as informações contidas na mensagem XML para registro de um serviço de armazenamento no lado esquerdo, distribuídas no modelo de dados do UDDI no lado direito. É possível visualizar também o uso de outros componentes do UDDI, como o já citado tModel, que facilita a pesquisa.

Seguindo a estrutura dos componentes do UDDI citados na subseção 2.2, e comparando com a Figura 2 vê-se o mapeamento dos seguintes componentes:

- *Business Entity* - Neste componente serão guardadas informações sobre o Serviço de Publicação e Descoberta, pois num cenário de múltiplos domínios será necessário diferenciá-los. É possível aqui atribuir um nome e características desejadas a uma entidade proprietária de um grupo de serviços. Sendo assim podemos ter a diferenciação entre o Serviço de Publicação e Descoberta da Rede Nacional de Ensino e Pesquisa (RNP) e o Serviço de Publicação e Descoberta da Géant por exemplo.
- *Business Service* - Serão armazenadas informações genéricas sobre o serviço

⁴GGF Global Grid Forum - www.gridforum.org

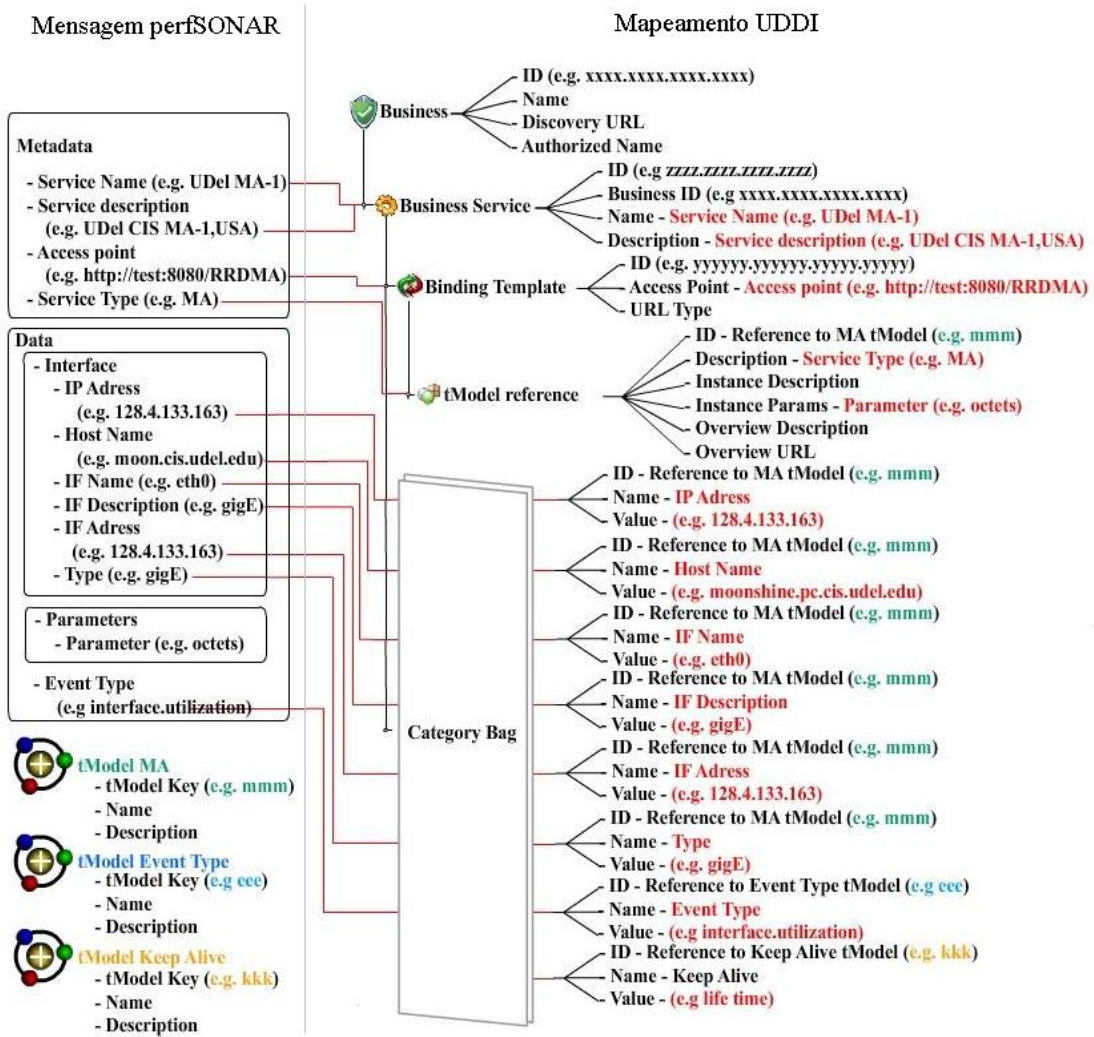


Figura 2. Mapeamento da mensagem de requisição do perfSONAR para o modelo de dados do UDDI.

como: nome, descrição e sua identificação no UDDI. Assim como no perfSONAR, as informações genéricas sobre o serviço não estão associadas a um identificador (taxonomia). Isso faz com que numa busca seja necessário obter todas as informações genéricas de todos os Serviços Web e, posteriormente, se faça outra busca nessa lista para localizar um determinado serviço. Essa associação não é feita pelo fato de poder existir serviços homônimos. Por isso as associações com índice só serão feitas em informações específicas.

- *Binding Template* - Guarda informações técnicas do serviço, as quais também podem ser usadas nas taxonomias, fazendo com que a busca seja baseada em uma característica. É esse componente também que guarda as informações de acesso do serviço com a URL de acesso e os parâmetros. No exemplo da Figura 2 é possível ver um componente chamado “tModel reference”. Com esse componente podemos fazer referência a um tModel que representa uma determinada taxonomia. Nesse caso faz-se referência ao tipo do serviço “MA”.
- *Category Bag* - Armazena outras informações consideradas mais específicas que

podem variar pelo tipo do serviço, e que pelo fato de serem específicas também são usadas como referência para taxonomias para categorizar o serviço.

Como visto na Figura 2, para armazenar as informações contidas no objeto *Data*, foi utilizado o componente “*Category Bag*”, que faz as ligações das taxonomias com o componente de identificação, o tModel, e ao mesmo tempo para armazenar informações específicas de um serviço que diferem das informações comuns. Como a mensagem XML do perfSONAR pode conter mais de um objeto *Data*, devido ao fato de um dispositivo onde são efetuadas medições poder dispor de mais de uma interface, o *Category Bag* também pode conter inúmeras informações ou até mesmo mais de um objeto *Category Bag*. Outro componente utilizado no UDDI foi o tModel reference que faz referência à taxonomia (tModel) principal que categoriza o serviço como do tipo MA (*Measurement Archive*) [Hanemann et al. 2005], neste exemplo. Com essa referência é possível achar, por exemplo, qualquer serviço de acordo com sua característica principal.

Para caracterizar os serviços de medição e assim poder criar uma taxonomia para esse serviço, foi necessário assimilar o conhecimento do trabalho realizado pelo grupo NMWG na categorização de dados de medição e padronização de mensagens. Logo, foi verificado que algumas informações contidas nas mensagens são importantes para a localização ou caracterização de um serviço, sendo elas: o tipo do serviço e a característica (*Event Type*). Outras informações também são importantes, mas eles só ajudam num refinamento maior da pesquisa. Com a definição dessas duas principais informações foram criadas duas taxonomias que as representassem, sendo a primeira para caracterizar o tipo de serviço e a segunda para caracterizar o *Event Type* do mesmo.

O NMWG descreve um grupo de características de rede e as classifica em uma hierarquia [Lowekamp et al. 2003], sendo possível identificar um serviço que fornece dados para uma determinada característica através dessa hierarquia. Como exemplo é possível citar a característica *delay.oneway*, um serviço atrelado à mesma significa que ele fornece dados de medição de atraso unidirecional. Um exemplo dessa taxonomia pode ser visto na Figura 3.

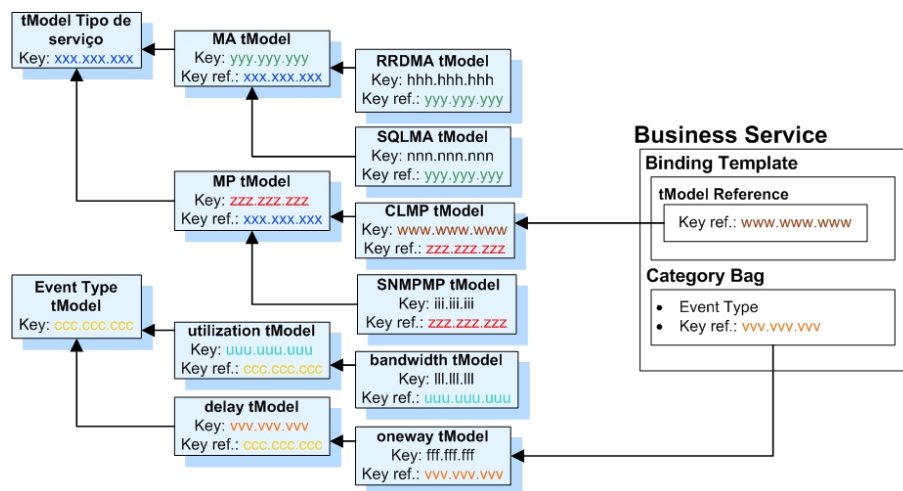


Figura 3. Principais taxonomias criadas para o modelo de dados de medição.

A outra taxonomia foi baseada no tipo do serviço. Logo, foi criada uma hierarquia

entre tipo de serviço e subtipo. Como exemplo é possível citar o tipo de serviço MP (*Measurement Point*), que possui vários subtipos como CL-MP (*Command Line Measurement Point*). Logo, com essa estrutura é possível identificar tipos específicos de serviços ou até mesmo um grupo dos mesmos. Essa taxonomia também pode ser vista na Figura 3.

Concluindo, essa estrutura definida possibilita várias formas de pesquisa para um Serviço Web. O usuário pode usar os índices criados automaticamente pelo UDDI para o serviço, bem como pode acessar os já existentes nas taxonomias ou localizar o serviço fazendo uma pesquisa no conjunto de dados através de uma informação específica.

4. Implementação realizada

O grupo de trabalho de medições da RNP, o GT-Medições [Monteiro 2005] vem pesquisando o desenvolvimento e a implantação de uma infra-estrutura de serviços de medição na rede. No início dos esforços da pesquisa, foi desenvolvida a arquitetura piPEs-BR [Sampaio et al. 2006], que consiste em um ambiente distribuído para medições composto por diversos elementos e processos de medição distribuídos potencialmente ao longo de diversos domínios administrativos. Essa infra-estrutura visa contemplar funcionalidades de: testes, armazenamento, agendamento, autorização, interface, publicação e descoberta de serviços, além de detecção/aconselhamento.

Um dos componentes da infra-estrutura piPEs-BR é o Serviço de Publicação e Descoberta, que em nossos estudos é baseado no uso das especificações da tecnologia UDDI. Como implementação dessa especificação foi utilizado o JUDDI, que é uma implementação *open source* em JAVA a especificação do UDDI, desenvolvida pelo grupo *Apache Software Foundation*. Essa ferramenta implementa a versão 2 da UDDI, que atualmente se encontra na versão 3. O JUDDI fornece as principais funcionalidades da tecnologia UDDI, faltando apenas a questão do uso de múltiplos serviços de UDDI, o que é imprescindível para uma infra-estrutura de monitoramento de multidomínios.

A vantagem do JUDDI é que, apesar da sua comunicação com seus usuários ser feita via Serviços Web, o mesmo armazena as informações sobre os serviços em uma base de dados relacional, o que resulta em um melhor desempenho em relação a uma base XML, por exemplo. Outra vantagem fundamental é que a JUDDI já possui um esforço em seu desenvolvimento e possui várias versões, significando que essa implementação já passou por várias fases, as quais um novo desenvolvimento ainda terá que passar.

Como parte integrante do Serviço de Publicação e Descoberta, foi desenvolvida uma biblioteca de acesso ao UDDI com o intuito de agilizar e facilitar o acesso ao serviço para os clientes, pois não é necessário que os usuários tenham conhecimento do funcionamento das APIs do UDDI. Essa biblioteca foi denominada de DPS-lib (*Discovery and Publish Service library*) e desenvolvida em JAVA, linguagem utilizada no desenvolvimento de toda a arquitetura do piPEs-BR. A idéia é que todos os usuários de serviços do piPEs-BR utilizem a biblioteca (Figura 4) para localizar os serviços e, posteriormente, acessá-los normalmente. A biblioteca possui métodos e objetos de acordo com as mensagens definidas pelo NMWG, ou seja, há objetos do tipo *Metadata* e *Data* como nas mensagens XML. Dessa forma é possível manipular as informações sobre os serviços de uma forma mais fácil, já que essa operação é feita através de métodos de acesso.

Outro desenvolvimento realizado nessa pesquisa foi a adaptação do Serviço de Publicação e Descoberta do perfSONAR, o LS, para utilizar a biblioteca desenvolvida e

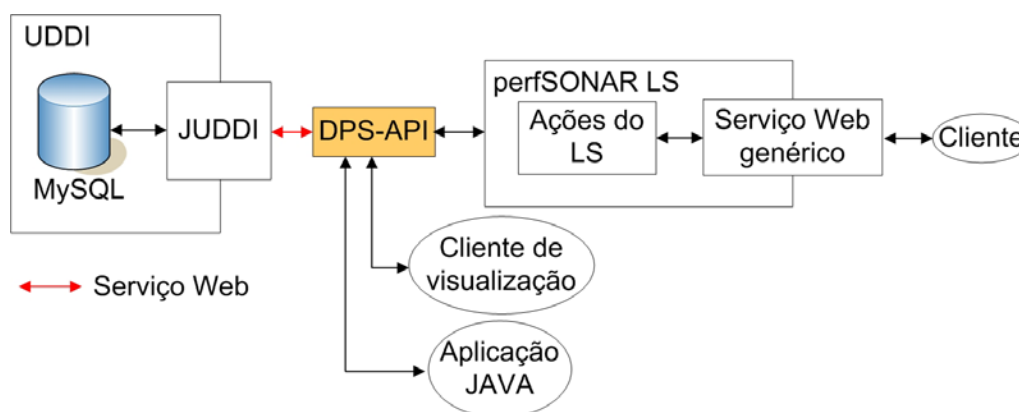


Figura 4. Arquitetura do Lookup Service na sua versão com UDDI.

interoperar com o UDDI. Esse desenvolvimento é justificado pela necessidade de tornar os *frameworks* perfSONAR e piPEs-BR interoperáveis e permitir que os dois serviços de Publicação e Descoberta sejam transparentes para o cliente, fazendo com que o mesmo utilize os dois sistemas da mesma forma. A adaptação do LS para comunicação com o UDDI é mais uma prova de que essa tecnologia pode ser usada em diversos contextos. Contudo, o cenário do Serviço de Publicação e Descoberta com o uso do UDDI pode ser utilizado de duas formas:

- Uma forma “universal” onde o cliente segue o padrão definido internacionalmente pelo NMWG e envia mensagens de requisição XML com as ações desejadas e recebe outra mensagem com o resultado. É importante dizer que para o cliente a implementação utilizada é transparente, já que a mesma pode ser com UDDI ou com a base XML.
- Um programa desenvolvido em JAVA utiliza a biblioteca DPS e acessa diretamente o serviço com UDDI.

Essas duas possibilidades de uso podem ser vistas na Figura 4, onde vemos o componente “Ações do LS” interoperando com a biblioteca. Com essa abordagem, tanto os componentes desenvolvidos no perfSONAR quanto no piPEs-BR podem utilizar o serviço, além de outros componentes que podem fazer uso da biblioteca diretamente sem precisar conhecer sobre as mensagens XML pré-definidas, já que a mesma disponibiliza acesso a métodos e objetos encapsulados em JAVA.

5. Testes e avaliação de desempenho

Para validar o Serviço de Publicação e Descoberta proposto neste artigo, além da implementação foram realizados testes com o serviço e com o *Lookup Service* (LS) que faz uso da base XML. Esses testes têm como objetivo fazer uma comparação entre as duas implementações e verificar qual possui o melhor desempenho.

Os testes foram realizados através de uma *benchmark* desenvolvida também como resultado desse estudo implementada na linguagem JAVA. Essa *benchmark* é um cliente do Serviço de Publicação e Descoberta onde podem ser escolhidas a quantidade de requisições e de ações a serem realizadas. A codificação da *benchmark* é baseada na função do JAVA “*System.currentTimeMillis()*.” e consiste em marcar dois intervalos de

tempo (inicial e final), contabilizando ao final a diferença entre os testes. O local de marcação do tempo inicial condiz com um instante antes da chamada do Serviço Web e o momento final logo após o recebimento da mensagem de resposta.

O cenário de teste pode ser visualizado na Figura 5, onde temos o cliente de testes localizado no prédio da Universidade Salvador (UNIFACS) que possui um link com o POP-BA da RNP. Já os dois serviços de *Lookup* foram instalados em um servidor dentro do POP-BA.

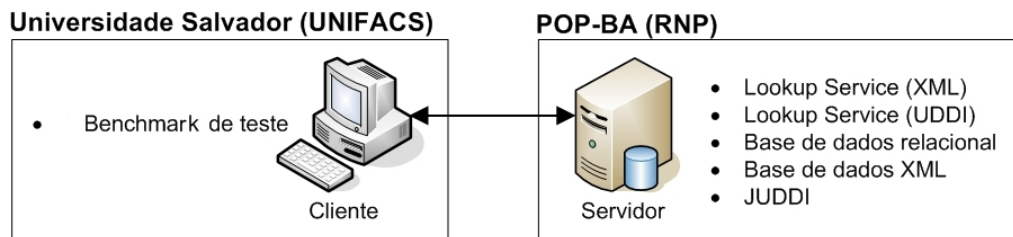


Figura 5. Cenário de testes.

Os testes foram realizados solicitando as seguintes ações para os serviços de Publicação e Descoberta: registro, alteração e deregistro. Foram usadas nos testes mensagens com informações sobre o serviço de armazenamento de dados (*Measurement Archive - MA*), o qual nesse exemplo armazena dados de utilização de interfaces que foram medidas. Essas mensagens contêm informações de acesso no *Metadata* (conforme relatado na seção 3) e armazenam também informações sobre cada interface onde estão sendo feitas coletas de dados através de objetos *Data*. Nestes testes foram usadas requisições com variação na quantidade de interfaces de 1 a 1.000, ou seja, documento XML com um *Metadata* de 1 a 1000 objetos *Data*. Os resultados podem ser vistos na Figura 6.

É possível observar que a solução usando o UDDI superou o serviço com base de dados XML à medida que o número de interfaces foi incrementado. Isso se dá devido ao fato do UDDI usar índices para registrar os serviços, além de não tratar os objetos *Data* como componentes separados e sim como tuplas de uma tabela que simplesmente podem ser inseridas. Essas vantagens são conhecidas, pois como já dito neste artigo, o JUDDI usa uma base de dados relacional. Já a base de dados XML não usa esquema de indexação o que dificulta a manipulação e faz com que os componentes fiquem esparsos.

6. Conclusão e trabalhos futuros

Com o estudo realizado foi demonstrada a possibilidade de se usar a tecnologia UDDI voltada para outros cenários além do *e-business*, e especificamente, na área de serviços de medição, informando dados de acesso e até informações específicas sobre medidas de redes. Esse estudo demonstra também que a tese na especificação do UDDI, que informa que o mesmo é um Serviço de Publicação e Descoberta de Serviços Web quaisquer é verdadeira, já que com uma modelagem da sua estrutura é possível criar um serviço funcional.

Conforme apresentado anteriormente, a ferramenta usada para a implementação da tecnologia UDDI foi a JUDDI. Como trabalho futuro é possível verificar a versão 3 da tecnologia do UDDI que é voltada para múltiplos domínios, o que descentraliza o

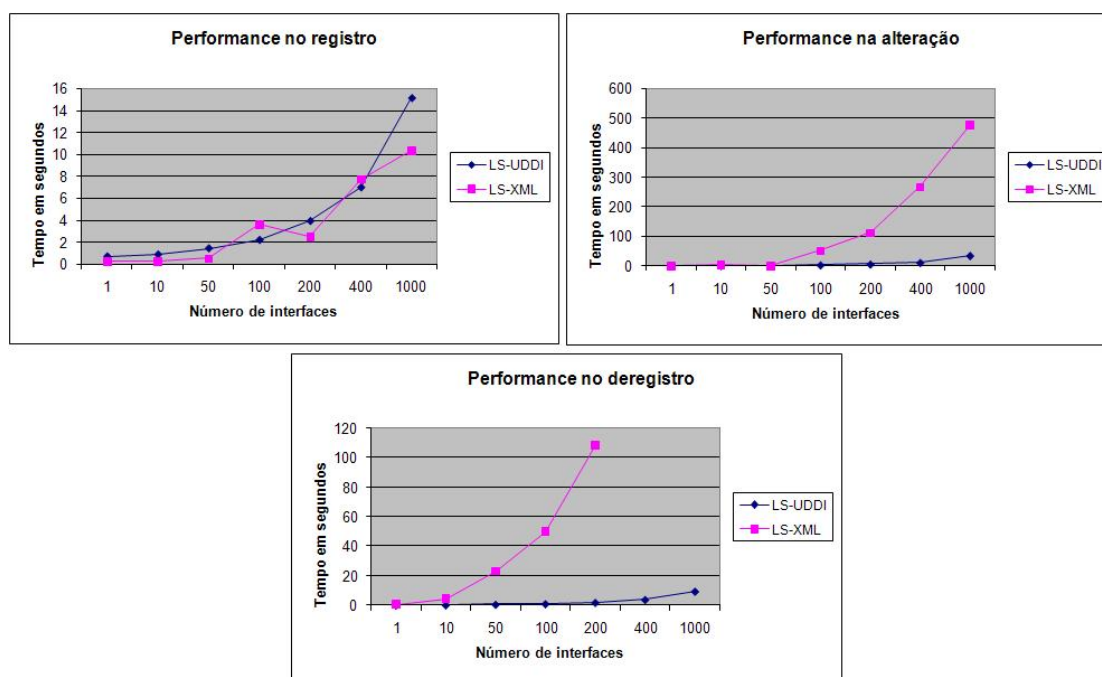


Figura 6. Desempenho do serviços de Lookup com base XML e com a integração com o UDDI.

serviço e o torna mais flexível e interoperável. A funcionalidade de múltiplos serviços é imprescindível quando se trata de uma ferramenta de multi-domínios, pois serviços de domínios diferentes podem trocar informações distribuindo os dados e sem a necessidade de que um serviço tenha conhecimento dos dados armazenados no outro. A versão 3 do UDDI aborda outras questões interessantes e de grande ajuda que são questões de segurança, onde o serviço pode atribuir níveis de acessos aos dados. O uso da versão 3 disponibilizaria mais funcionalidades, mais confiabilidade e grande abrangência.

Outro trabalho que pode ser desenvolvido com base nos estudos feitos nesse artigo é o uso de outras tecnologias em conjunto com o UDDI para melhorar ainda mais o uso dos Serviços Web da infra-estrutura. Podemos citar como exemplo o uso da Web Semântica [Paolucci et al. 2002] no UDDI. Com ela é possível atribuir significado nas pesquisas no UDDI e realizar as mesmas de forma mais refinada, fazendo ligações entre as informações armazenadas e correlacionadas aos serviços de modo que é possível descobrir um ou mais serviços para um determinado grupo de características. É possível também neste conjunto de novas funcionalidades, utilizar junto com o UDDI a composição de serviços, onde o Serviço de Publicação e Descoberta pode compor um novo serviço a partir de dois ou mais serviços e fornecer essa informação para o cliente. Uma tecnologia que pode ser usada e que sobre a qual já existem estudos é o BPEL (*Business Process Execution Language*) [Riegen and Trickovic 2004] com UDDI.

Agradecimentos

Os autores agradecem a Fundação de Amparo a Pesquisa da Bahia (FAPESB), a Rede Nacional de Pesquisa (RNP) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio dado para a realização do trabalho.

Referências

- Apte, N., Mehta, T., et al. (2002). *UDDI: Building Registry-based Web Services Solutions*. HP Professional Series. Prentice Hall.
- Berwanger, R. et al. (2001). Message service specification: ebxml transport, routing & packaging. <http://www.ebxml.org/specs/ebMS.pdf>.
- Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002). Unraveling the web services web: an introduction to soap, wsdl, and uddi. *IEEE Internet Computing*, pages 86 – 93.
- Ehnebuske, D., McKee, B., Rogers, D., et al. (2002). Uddi version 2.04 api specification. http://uddi.org/pubs/ProgrammersAPI_v2.pdf. UDDI Published Specification.
- Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., and Nielsen, H. F. (2003). Soap version 1.2 part 1: Messaging framework. <http://www.w3.org/TR/soap12-part1>. W3C Recommendation.
- Gudgin, M., Hadley, M., Moreau, J.-J., and Nielsen, H. F. (2001). Soap version 1.2 part 1: Messaging framework. <http://www.w3.org/TR/2001/WD-soap12-part1-20011002>.
- Hanemann, A., Boote, J. W., Boyd, E. L., Durand, J., Kudarimoti, L., Lapacz, R., Swany, D. M., Trocha, S., and Zurawski, J. (2005). Perfsonar: A service oriented architecture for multi-domain network monitoring. *In Proceedings of the Third International Conference on Service Oriented Computing (ICSOC 2005)*. ACM Sigsoft and Sigweb.
- Lowekamp, B., Tierney, B., Cottrell, L., Hughes-Jones, R., Kielmann, T., and Swany, M. (2003). A hierarchy of network performance characteristics for grid applications and services. <http://nmwg.internet2.edu/docs/nmwg-measurements-v14.pdf>.
- Meier, W. (2002). exist: An open source native xml database. *Computer Science*. Wolfgang Meier. eXist: An Open Source Native XML Database. In Erhard Rahm B. Chaudri, Mario Jeckle and Rainer Unland, editors, *Web, Web-Services, and Database Systems*, 2593, Erfurt, Germany, 2002. Springer LNCS Series.
- Monteiro, J. A. S. (2005). Gt-medições 2: Proposta para grupo de trabalho. Technical report, Rede Nacional de Pesquisa - RNP.
- Paolucci, M., abd Terry R. Payne, T. K., and Sycara, K. P. (2002). Importing the semantic web in uddi. *In International Workshop on Web Services, E-Business, and the Semantic Web*, pages 225 – 236.
- Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. *In Keynote for the 4th International Conference on Web Information Systems Engineering (WISE 2003)*. IEEE Computer Society.
- Riegen, C. V. and Trickovic, I. (2004). Using bpel4ws in a uddi registry. <http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-bpel.htm>.
- Roy, J. and Ramanujan, A. (2001). Understanding web services. *IT Professional*, pages 69 – 73.
- Sampaio, L., Koga, I. K., Souza, H., Koga, I. K., Rhoden, G. E., Vetter, F., Leiria, G., Monteiro, J. A. S., and Melo, E. (2006). pipes-br: Uma arquitetura para a medição de desempenho em redes ip. *In XXIV Simpósio Brasileiro de Redes de Computadores*.